U.S. Department of the Interior
U.S. Geological Survey

The National Map Catalog Technical Discussion Paper

The National Map Catalog Service: A Guide for Application Developers

Document version 0.2.0
For catalog service versions 1.1.0 and 2.0.0

Jeff Wendel
Larry Moore

Mid-Continent Mapping Center
September 2004

Table of Contents

1 Introduction

The National Map manages a database that tracks information about Web Map Services (WMS). This database can be queried through a web application running in a servlet container on a USGS computer. This web application is a **catalog service** that is consistent with the Open GIS Consortium (OGC)[1] Draft Candidate Specification "OGC Web Services Stateless Catalog Profile."

The National Map catalog is therefore composed of two parts: a database and a service for accessing the database. The database is implemented in Oracle, and is not accessible outside the USGS. The associated service allows any application anywhere on the Internet to extract information from the database.

Requests to the catalog service are in http syntax. Requests are typically sent by software applications, but can also be interactively entered in a browser address window. Responses are in Extensible Markup Language (XML) document type definition (DTD) form. The XML responses must be interpreted by the client program. Because XML is human-readable, requests typed into a browser address window will return documents that can be read in the browser. This is useful for training, but is of limited practical value for real work.

The catalog service is a stateless application, meaning there is no concept of a session.

1.1 Notes on Version Numbers

The first documented release of the catalog service was numbered 1.1.0. By unfortunate coincidence, this is the same version number as one of the important versions of the OGC WMS Specification. The confusion this would cause was not appreciated until it was too late to do anything about it. There is actually no relationship between the version of the WMS specification and the version of the catalog service.

This paper and its predecessors document that catalog service. The version numbers of these documents do not match the version numbers of the software they describe; they are related more closely to the authors' subjective ideas about the maturity of the documentation. Document versions 0.1.0 through 0.1.3 describe catalog service version 1.1.0. Document versions 0.2.x describe catalog service version 2.x.x.

1.2 Notes on Formal Standards Processes

The catalog service was originally designed to be consistent with the Open GIS Consortium (OGC) discussion paper "Web Registry Services," later renamed Draft Candidate Specification "OGC Web Services Stateless Catalog Profile." Version 0.06 of this draft specification is posted on The National Map catalog Web site. In the introduction, this document states an intention to "…integrate this document with the OGC Catalog Services v 1.0…"

The OpenGIS Catalog Services Specification has now replaced the Web Services Stateless Catalog Profile, but the two were in parallel development for several years. In August 2004 version 2.0 of the Catalog Services Specification (OGC 04-021r2) was released.

[1] At about the time this document was being finalized, the Open GIS Consortium changed its name to the Open Geospatial Consortium. The older term is used throughout this document, and will be corrected in the next version. The acronym OGC is unaffected by the name change.

The USGS intends to implement *The National Map* catalog in conformance with the OGC Catalog Services Specification. This is not a simple ambition, because of the evolving nature of the various standards and systems involved:

- The catalog service is a working software system that existing applications depend on to function properly. It is absolutely critical that changes and enhancements to the catalog service interface do not break features of the service in previous software versions. All new versions **must** be backwardly compatible with earlier versions.
- The OGC Catalog Services Specification has been evolving and expanding at a relatively rapid rate in recent years. The most recent version is so new that it is not clear whether this period of standard definition is over or not.
- Functional requirements for the catalog service continue to evolve in response to political and technical demands within the USGS and from *The National Map* business partners. These requirements are more or less independent of formal standards processes.

At this time (fall 2004) we believe the catalog service implementation is consistent **in spirit** with the OGC Catalog Services Specification, and that in time – assuming this project lives and prospers – true conformance can be achieved. The current implementation is probably technically inconsistent with the OGC specification, and is certainly incomplete.

As a separate but related issue, the USGS has its own product standards, and is currently working on a family of standards that will describe *The National Map*. This documentation of the catalog service is proposed to be one such standard, and this document is currently being rewritten to conform to the appropriate USGS editorial guidelines. When that process is complete, this document will be replaced by a more formal and stable document. The new standard will have essentially the same content, but will have a different appearance and probably a different version numbering convention.

1.3 Catalog Service Version 2.0.0

The first documented version of the catalog service was 1.1.0, released in March 2004. *The National Map* is still in a relatively early stage of development, and requirements for all parts of the system continue to evolve. In August 2004 a significantly different version of the catalog service, version 2.0.0 was released.

Version 2.0.0 is backwardly compatible with version 1.1.0. This means that existing applications based on 1.1.0 will continue to work. Upgrading applications to 2.0.0, while relatively easy, is not completely painless. An application that uses the additional features available in 2.0.0 must explicitly request that version in its requests to the catalog service. Applications written in Java or other procedural languages may require syntax changes to some function calls. See the software release notes for more details.

The differences between the two versions are summarized in Attachment C of this document, noted throughout the body of the document, and are also summarized in the software release notes.

2 Hosts

At this writing, the catalog service runs concurrently on two hosts. The primary host, from the perspective of USGS-owned applications of *The National Map*, is

helios.er.usgs.gov:8080

Helios is physically located at Mid-Continent Mapping Center. When helios is down, the USGS applications switch to

t204dwimdn.er.usgs.gov

This machine is physically located at a USGS Water Resources Discipline (WRD) office in Wisconsin.

The two instances of the service are synchronized, so an application can use either at any time.

3 General Form of Catalog Service Requests

A request to the catalog service has the general form

http://<host name>:[port]/ <servlet path>?[query string]

where

[] denotes 0 or 1 occurances

<> denotes exactly 1 occurance

For example:

http://helios.er.usgs.gov:8080/catprod/start?request=getRecords&
querySpec='bbox=-96,35,-95,36|classifications=roads|ranking=NONE|
allClassifications=FALSE|scale=.0001|'

In this example:

- host name = *helios.er.usgs.gov*
- port = *8080*
- servlet path = *catprod/start*
- query string is

request=getRecords&
querySpec='bbox=-96,35,-95,36|
classifications=roads|
ranking=NONE|
allClassifications=FALSE|
scale=.0001|'

(line breaks and indents are inserted in this example to improve readability).

The domain name, port, and servlet path parts of the request are straight-forward, but the query string part has a complex structure of its own.

4 Query Strings

Query strings are technically optional, but a request with no query string returns a small amount of connection information and therefore has very limited value.

A query string has the form

request=requestType[&**querySpec**='{value|}'] [&**version**=value]

where

[] denotes 0 or 1 occurances

{} denotes 0 or more occurances

requestType and *value* represent variables. A *requestType* must be from the list of seven possible types below. Depending on context, *value* may have a continuous range or may take only specific defined values.

request, *querySpec*, and *version* are string literals.

version may take only two values: 1.1.0 or 2.0.0. If *version* is not present, a value of 1.1.0 is assumed.

The four symbols in the set [= ' | &] are string literals (the square brackets [] are not themselves part of this set)

Query strings are case-insensitive.

4.1 requestType

Version 1.1.0 of the catalog service supports three requestTypes:

1. getCapabilities
2. describeRecord
3. getRecords

Version 2.0.0 of the catalog service supports four additional requestTypes:

4. getClassifications
5. getElementSets
6. getElementStatuses
7. getServiceStatuses

4.1.1 getCapabilities

A getCapabilities request is used to gather metadata about the catalog service itself:

http://helios.er.usgs.gov:8080/catprod/start?request=getCapabilities

The response to this request is an OGC-compliant capabilities XML document that describes the capabilities of the catalog service (as opposed to the capabilities of some WMS, which would be a more common example of a capabilities document). The catalog service capabilities document is currently a static file that contains mainly contact and access URL information.

4.1.2 describeRecord

A describeRecord request will return a DTD describing the format of a getRecords response:

http://helios.er.usgs.gov:8080/catprod/start?request=describeRecord

http://helios.er.usgs.gov:8080/catprod/start?request=describeRecord&version=2.0.0

The DTD returned by this request contains an embedded data dictionary (in comment fields) that describe the data elements of the catalog service.

4.1.3 getRecords

The most important and complex requestType is getRecords. In a sense, the getRecords request is the purpose of the entire service, and everything else exists to support it. This request can be issued with or without a querySpec parameter, though nearly all normal uses of the request will include a querySpec to constrain the request. If the querySpec parameter is omitted, the request is essentially an unconstrained query for everything the service can report about the database:

http://helios.er.usgs.gov:8080/catprod/start?request=getRecords

This request (with no querySpec parameter) will return a very large XML response containing information for all public services and all public layers. This is nominally an unconstrained query, but because some querySpec parameters have defaults, it returns information only for services and layer that are (1) available, (2) public, and (3) part of *The National Map* element set. To return information about all services and layers, these constraint must be explicitly overridden with a querySpec parameter.

An example of a more normal getRecords request, with a querySpec parameter, is given in section 3.

4.1.4 getClassifications (version 2.x only)

Returns a classification list – that is, a list of subthemes. getClassifications may be constrained with an optional querySpec.

Classifications allow requests such as "what layers are available that show open-water features in this area?"

4.1.5 getElementSets (version 2.x only)

Returns a list of all element sets that contain at least one element:

http://helios.er.usgs.gov:8080/catprod/start?request=getElementSets&version=2.0.0

This request cannot be constrained with a querySpec.

Element sets support customized applications. The catalog database contains information about a large and growing number of WMSs. Though the database's primary purpose is to support applications of the USGS's *The National Map*, it is easy to see how this inventory of data sources could be useful for other applications. But not all applications are interested in exactly the same datasets. For example, an application designed by and for State X will not be interested in datasets that only cover State Y.

More importantly, some applications may need data that should not be visible to other applications. A general public viewer and a Department of Homeland Security (DHS) disaster-relief application will both need certain base data layers, but the DHS application may also depend on data layers that should not be available to the general public.

Element sets are an elegant solution to this problem. Elements (WMS layers) can be grouped together in the catalog database in sets. Applications that know what element sets they are interested in can access only those sets, and ignore all other sets. This does not, by itself, provide security for sensitive data layers. Actual data security must be provided by the services themselves through password protection or encryption. But grouping layers into sets allows cooperating

applications to show only what they are interested in and avoid confusing login screens and "access denied" messages from other services.

4.1.6 getElementStatuses (version 2.x only)

Returns a status list containing the domain of element statuses.

http://helios.er.usgs.gov:8080/catprod/start?request=getElementStatuses&version=2.0.0

This request cannot be constrained with a querySpec.

Status is an attribute of both services and elements. The field is used as a hint to applications about using the service or element. Examples of status:

- A value of PUBLIC for both an element and the service it comes from means this GIS layer is within the domain of *The National Map*, has been evaluated by the USGS, is believed to be accurate and important, and is not highly duplicative of other PUBLIC layers.
- REVIEW means the service or element is being considered for inclusion in *The National Map* element set, but is still under evaluation. In some cases the element may be lacking some characteristic (such as associated geospatial metadata) required by *The National Map*, but is expected to acquire that characteristic soon.
- NOT USED means the service or element has been evaluated and is considered inappropriate for or outside the scope of *The National Map*. There are many reasons for an element to be marked this way, some of them having nothing to do with quality or completeness. NOT USED does not necessarily mean there is anything wrong with the data.

The status attribute is a simple text string. The domain of possible values is currently not limited.

4.1.7 getServiceStatuses (version 2.x only)

getServiceStatuses – Returns a status list containing the domain of service statuses. This request cannot be constrained with a querySpec.

Though not required to have the same domain of values, service status and element status are comparable. Status is an attribute of both element and service primarily to allow all the layers in a service to be flagged together with a particular status.

4.2 querySpec

Two request types, getRecords and getClassifications, can be constrained with an optional querySpec. Constraining these requests is a critical feature of the catalog service, particularly for the getRecords request. Without this ability to impose constraints, the service would be of very limited value to GIS applications, because all requests would return a huge set of responses.

A querySpec for both the getRecords and getClassifications requests has the following form, and may use any number of these parameters:

querySpec='
bbox=<minX,minY,maxX,maxY>|
classifications=<classificationList>|
scale=<scaleFactor>|

```
allClassifications=[TRUE|FALSE]|
ranking=[NONE|SCALECLASS]|
element=<elementID>
elementset=<elementSetID>|
servicestatuses=<serviceStatuses>|
elementStatuses=<elementStatuses|
serviceavailable=[T|F|ALL]|
'
```

(line breaks and indents are inserted in this example to improve readability).

Where:

- [] indicates that exactly one of the listed options is used.
- minX,minY,maxX,maxY define a bounding box constraint in WGS84 geographic decimal degrees.
- classificationList is a comma separated list of subthemes of *The National Map*. No whitespace should exist before or after a list member.
- scaleFactor is a value expressed in decimal degrees/pixel.
- elementID is the identification number (primary key) of one element.
- elementSetID is the identification number (primary key) of the requested catalog element set
- serviceStatuses is a comma-delimited list of status codes. No whitespace should exist before or after a list member.
- A querySpec parameter list is enclosed in single quotes and consists of parameters ending and separated by the ' | ' character.

The constraints of querySpec are strung together as 'and' conditions. For example, a getRecords querySpec that includes the constraints "classifications=roads|scale=.0001" will return only records that are classified as roads AND that have a maxviewscale=.0001 and a minviewscale=.0001.

4.2.1 bbox

The bbox parameter is used to spatially constrain the query. Internally, the catalog associates WGS84 geographic polygonal footprints with each layer. The catalog service uses these stored footprints to determine intersection with the bbox and return information about layers where such an intersection occurs.

The bbox is defined with coordinates in this order: minX,minY,maxX,maxY, where all values are WGS84 geographic decimal degrees, west negative. In the northwest quadrant of the globe, the first X,Y pair correspond to the southwest corner of the box and the second X,Y pair correspond to the northeast corner.

4.2.2 classifications

The classifications parameter is used to constrain the query to selected data subthemes. *The National Map* categorizes all data by theme and subtheme – “classification” is the term used in the database for subtheme. An example of a theme is hydrography. Classifications in the hydrography theme include steams, lakes, and wetlands.

Version 1.1.0 contains no efficient method to dynamically obtain the classifications list. This shortcoming removed in version 2.0.0 with the addition of the getClassifications requestType (see section 4.1.4).

4.2.3 scale

Every element (GIS layer) in the catalog has assigned min and max viewscales. These values are hints to applications about the appropriate range of display scales. The scale parameter means "return a layer only if this *scale* parameter is between minviewscale and maxviewscale inclusive."

Viewscales and this scale parameter are expressed in decimal degrees per screen pixel. For a detailed discussion of viewscales, see the paper "Viewscales and their Effect on Data Display" at http://mcmcweb.er.usgs.gov/catalog/tnm_catalog_page2.html#techinfo.

4.2.4 allClassifications

This is a deprecated parameter. Though it works as described, it is unlikely to be terribly useful.

The allClassifications parameter partially overrides constraints on classifications. If allClassifications=TRUE then all are included, but layers not classified with a member of the classifications parameter will have a ‘requested=false’ attribute in the response. Layers that are classified with a member of the classifications parameter will have an attribute of 'requested=true' in the response. For example, the querySpec fragment

classifications=ROADS,STREAM NETWORK|allClassifications=TRUE|

will return records for layers with any classification, but attribute those with a classification of ROADS or STREAM NETWORK with requested=true and all others with requested=false. If allClassifications is FALSE or absent, then only those layers with a classification in the classifications parameter will be returned in the response and every layer will be attributed with requested=true.

4.2.5 ranking

The ranking parameter determines how the response layer attribute 'recommended' is populated. If this parameter is absent or ranking=NONE, then all layers will have the attribute 'recommended=true'. If ranking=SCALECLASS then only the largest-scale layers in each classification necessary to completely fill the bbox will have the attribute ‘recommended=true’, the rest will have the attribute ‘recommended=false’. It is recommended ranking only be used by advanced clients with a good understanding of its behavior.

4.2.6 element

Specifies by element ID number which elements should be returned.

4.2.7 elementSet

The elementSet parameter determines which element sets to consider. The parameter is the numerical database key of an element set. *The National Map* element set id is 1 and is the default if this parameter is absent. A parameter value of 0 may be passed to consider all elements without regard to which element sets they belong.

4.2.8 serviceStatuses

The serviceStatuses parameter constrains the services to be considered to those with a status attribute in the comma delimited value list. A default value of PUBLIC is used if this parameter is absent. A value of ALL includes all services without regard to the status attribute.

The most common status values are PUBLIC (service is available to *The National Map* public viewer), REVIEW or NEW (data under review, may soon be available to *The National Map* public viewer), and NOT USED (data not available to *The National Map* public viewer). The complete domain of service statuses can be obtained with the getServiceStatuses requestType (see section 4.1.7).

4.2.9 elementStatuses

The elementstatuses parameter constrains the elements to be considered to those with a status attribute in the comma delimited parameter value list. A default value of PUBLIC is used if this parameter is absent. A value of ALL includes all elements without regard to the status attribute.

The complete domain of element statuses can be obtained with the getElementStatuses requestType (see section 4.1.6).

4.2.10 serviceAvailable

USGS-hosted software continually checks the availability of services registered in the catalog. When a service does not respond to a well-formed query, the service is marked as unavailable. Applications may wish to avoid querying services that are known by the catalog to be off-line. The serviceAvailable parameter constrains the services to be considered to those with an available attribute equal to the parameter value. A default value of T, for true, is used if this parameter is absent. A value of ALL includes all services without regard to the availability attribute.

5 References

Extensible Markup Language (XML) 1.0, 6-October-2000,
http://www.w3.org/TR/2000/REC-xml-20001006.pdf

OGC Web Services Stateless Catalog Profile, version 0.0.6, 29-August 2001.
http://mcmcweb.er.usgs.gov/catalog/docs/2001-062StatelessCatalogProfile.pdf

OpenGIS Catalog Services Specification, version 2.0, 11-May-2004
http://www.opengeospatial.org/specs/

The National Map catalog web site. http://mcmcweb.er.usgs.gov/catalog

Attachment A. Reading XML Responses with Software

The National Map viewer client is the first example of a web application that is driven by *The National Map* catalog service. The viewer software reads a getRecords response with a Java library generated with Java Architecture for XML Binding (JAXB), Early Access. Sample code from this application, and the JAXB EA jar file, can be retrieved from http://mcmcweb.er.usgs.gov/catalog/api/samples/catalog_web_app_examples.zip.

If the client is running in a servlet container, it is important that JAXB 1.0 (or later) not be loaded by the container when the container starts. If you are using the Tomcat container bundled with the JWSDP this can be accomplished by renaming the jaxb-X.X directory in your distribution to something like jaxb-X.X-dist. The getRecords reader and JAXB EA jar files should be placed in your web applications lib directory.

Attachment B. Example response

The service request

http://helios.er.usgs.gov:8080/catprod/start?request=getrecords

can be typed into a browser address window. It returns an XML document, which is a convenient format for machine parsing, but is also human-readable. This example is an unconstrained request for Catalog records. It essentially says "return all information about all services and all layers." The resulting XML document would be some 350 pages long if printed. The body of this paper describes how to constrain such requests to return more limited and focused record sets.

Following are some excerpts from the response to this request to illustrate the types of information that can be retrieved through the catalog service.

Service Information

For each service registered in the catalog, a block similar to the following is returned:

```
- <Service>
    <Version>1.1.1</Version>
    <Title>Tahoe Pilot WMS</Title>
    <Abstract />
    <Status>PUBLIC</Status>
    <PartnerName>Tahoe Partners</PartnerName>
    <PartnerWebURL>http://mapsonline.wr.usgs.gov/nm_tahoe_feature/
      metadata/Lake_Tahoe_Partners.html</PartnerWebURL>
    <GetCapabilitiesURL>http://mapsonline.wr.usgs.gov/ogcwms/servlet/
      com.esri.ogc.wms.WMSServlet?servicename=WMS_tahoe_pilot
      </GetCapabilitiesURL>
    <GetMapURL>http://mapsonline.wr.usgs.gov/ogcwms/servlet/
      com.esri.ogc.wms.WMSServlet?servicename=WMS_tahoe_pilot
      </GetMapURL>
    <GetFeatureInfoURL>http://mapsonline.wr.usgs.gov/ogcwms/servlet/
      com.esri.ogc.wms.WMSServlet?servicename=WMS_tahoe_pilot
      </GetFeatureInfoURL>
    <BackingURL>http://mapsonline.wr.usgs.gov/servlet/
      com.esri.esrimap.Esrimap?ServiceName=tahoe_pilot</BackingURL>
    <BackingService>ESRI-ArcIMS</BackingService>
    <BackingVersion>1.1</BackingVersion>
```

The name of this service is "Tahoe Pilot WMS." It is owned by the partner "Tahoe Partners." The service has status PUBLIC, meaning its data are available for display in *The National Map* viewer.

The URLs to the provider home page and the service itself are the most important pieces of information. These allow applications to connect to the service and retrieve its GIS data for display or analysis.

Layer Information

One service can have any number of GIS layers. Within the <Service> tag block are sub-blocks with information about each layer. For example:

```
- <Layer id="50" requested="true" recommended="true"
    transparency="false" queryable="false" downloadable="true">
    <Name>NLC_image</Name>
    <Title>Tahoe National Land Cover</Title>
    <PartnerName>Tahoe Partners</PartnerName>
    <PartnerWebURL>http://mapsonline.wr.usgs.gov/nm_tahoe_feature/
     metadata/Lake_Tahoe_Partners.html</PartnerWebURL>
    <Resolution>0</Resolution>
    <MinScale>0.00002141514865300 36</MinScale>
    <MaxScale>0.0048</MaxScale>
    <TimePeriod>01-jan-1900</TimePeriod>
  - <Classification id="7" zdepth="208010.00005195">
      <Name>LAND COVER</Name>
      <Theme>LAND USE/LAND COVER</Theme>
    </Classification>
    <SRS>54008</SRS>
    <SRS>4326</SRS>
    <LatLon>Undefined</LatLon>
    <BoundingBox>Undefined</BoundingBox>
    <Status>PUBLIC</Status>
    <Type>RASTER IMAGE</Type>
    <MetadataURL>http://landcover.usgs.gov/
      nationallandcover.html</MetadataURL>
    <LogoURL />
    <SLDURL />
    <LegendURL />
    <BackingName>NLC_image</BackingName>
    <Description>Image covering the project area showing the
      type of surface coverage at a resolution of 30 meters per
      pixel.</Description>
</Layer>
```

The “name” field is what the service calls this layer (NLC_image). The “title” field is populated by USGS personnel, and is used by USGS applications such as the viewer to describe this layer. In this case, the string “Tahoe National Land Cover” is used by the viewer to label this layer. Other information includes the min and max viewscales for the layer, the display status, metadata URL, and the spatial reference systems the service supports for this layer.

Much of this information is supplied by the WMS serving the data, and is stored in the USGS Catalog database through a semi-automated harvest process. However, some of the information, such as the viewscale values and metadata URL, are added and maintained by the USGS to assist *The National Map* applications.

Field definitions

The explanations of the two examples above are obviously not complete. A complete and formal data dictionary of the fields is needed. The documentation problem has two parts:

1. Most of the fields returned in catalog service responses correspond directly to a field in the catalog database. When the database data dictionary is fully populated, the definitions of these fields can be derived directly from the database. However, the names of the database fields are not exactly the same as the names of the service fields in all cases. Therefore, some cross-walk between field names will be needed.
2. In addition, the service returns some fields that are derived or calculated from information in the database. These field names therefore do not directly correspond to any database field, and their definitions will need to be maintained separately from the database.

Concurrent with the release of version 2.0.0 of the catalog service, data dictionary documentation was added to the catalog service DTD. Definitions of the catalog service fields can therefore be obtained with a describeRecord request:

http://helios.er.usgs.gov:8080/catprod/start?request=describeRecord

This documentation method is not completely satisfactory. It is one more static document that must be manually kept consistent with the database and other documentation. On the other hand, the DTD is a technically critical component of the overall system; since the DTD fields **must** be kept current and consistent, it should be relatively easy to keep the field definition comments up to date as well.

Questions about specific questions about the meanings of fields, or suggestions for improving the DTD documentation, can be sent to the Catalog Support Team at USGScatalog@usgs.gov.

Attachment C. Summary of Differences Between Catalog Service Versions 1.1.0 and 2.0.0

In any software system, adding new features while retaining backward compatibility creates problems and requires compromises. Through 2004 it was obvious that the early versions of the catalog service needed additional requestTypes.

Version 2.0.0 of the catalog service adds four requestTypes. Implementing these in a way that does not break any existing applications is both critically important and fairly difficult.

Version 2.0.0 of the catalog service implements the idea of a capability specification to clearly specify which version of the service an application intends to use. The default is the original 1.1.0 version, so if requests to the service do not specify a version, 1.1.0 is assumed. If an application wants to use version 2.0.0, the application must specify this within each request string. A specification is selected by adding

version=<specification version>

parameter to the query string (see section 4 in the report body). If this parameter is missing, the service assumes 1.1.0. Version 1.1.0 may also be specified explicitly.

New specifications always support the request types of previous versions and never remove elements or attributes from previous version response formats. Rather, new versions add request types and extend responses with optional elements and attributes. This allows applications to use the features of a new specification with little change to existing source code.

The 1.1.0 specification was described in version 0.1.2 of this paper. As described in section 4.1 of this document, version 1.1.0 defined three request parameter values:

1. getCapabilities – returns a capabilities document, essentially a description of what the catalog service can do.
2. describeRecord – returns the DTD of the catalog service.
3. getRecords – returns information contained in the catalog database about services and GIS layers.

Version 2.0.0 adds four new request types:

4. getClassifications – Returns a classification list (that is, a list of subthemes) constrained with an optional querySpec.
5. getElementSets – Returns an element set list. The list contains all element sets that contain at least one element. This request cannot be constrained with a querySpec.
6. getElementStatuses – Returns a Status list containing the domain of element statuses. This request cannot be constrained with a querySpec.
7. getServiceStatuses – Returns a Status list containing the domain of service statuses. This request cannot be constrained with a querySpec.

The following additions were made to the 2.0.0 response DTD:

1. A searchResults element can now contain exactly one Service, Classification, ElementSet or Status list.
2. An optional id attribute was added to the Theme element.